Claims

WHAT IS CLAIMED IS:

Sub A1

1. A computing system having a mass storage device and a system timer for obtaining benchmark timing for a portion of an application program execution, the computing system comprising:

a mass storage system;

an init module for determining if the timestamp data is to be collected during the operation of the application program;

a performance marker module for obtaining and storing the timestamp data for later retrieval;

an uninit module for formatting and storing the obtained timestamp data into a data file within the mass storage device that permits retrieval after the termination of the application program; and

a performance benchmark data post processing module for determining the benchmark timing from two or more timestamp data entries;

wherein the performance marker module is executed at predefined points within a plurality of processing modules within the application program.

Sub B2

2. The system according to claim 1, wherein:

the init module is executed before any timestamp data is collected;

the performance marker module is executed each time benchmark timestamp data and overhead timestamp data is to be collected;

the uninit module is executed after all timestamp data desired has been collected to store the timestamp data within records of a Raw Data Table; and the performance benchmark data post processing module determines the benchmark timing from the records stored within the Raw Data Table.

3. The computing system according to claim 2, wherein the init module determines if timestamp data is to be collected.

4. The computing system according to claim 3, wherein init module makes the determination that timestamp data is to be collected by checking for the existence of an identification key within a system registry;

the identification key uniquely identifies the processing modules to be used to collect, format, and store the run-time internal state data to be collected.

B2 cont

5. The computing system according to claim 4, wherein the performance marker module collects timestamp data only if the init module has determined that the timestamp data is to be collected.

6. The computing system according to claim 5, wherein the performance marker module generates a data record within the Raw Data Table each time the performance marker module is executed.

7. The computing system according to claim 6, wherein the benchmark data record further containing an overhead timestamp data value each time the performance marker module is executed.

8. The computing system according to claim 7, wherein the performance marker module stores the benchmark data records within a data memory block within the processing modules identified by an identification key within a system registry.

9. The computing system according to claim 8, wherein the uninit module retrieves the data records from the data memory block for transfer to the Raw Data Table on the mass storage device.

10. The computing system according to claim 9, wherein the performance benchmark data post processing module determines the benchmark timing from difference between two benchmark timestamp data entries stored within the Raw Data Table to generate a second data Record within a Processed Data Table.

Sub B2 cont

11. The computing system according to claim 10, wherein the data record within the Raw Data Table comprises a ResultID field, an AppID field, a MarkerID field, a Marker Cycles Field, and an Overhead Cycles field.

12. The computing system according to claim 11, wherein the second data record of the Processed Data Field comprises a ResultID field, a Reboot Iteration field, a Launch Iteration field, a Marker Iteration field, a Marker Pair ID field, and a Seconds Field.

13. The computing system according to claim 12, wherein the ResultID field of the Raw Data Table corresponds to the ResultID field in the Processed Data Table.

14. The computing system according to claim 12, wherein the Marker Pair ID field corresponds to a second Marker Pair ID field in a Marker Pair Table.

15. The computing system according to claim 14, wherein the Marker Pair Table comprises the second Marker Pair ID field, a start App ID field, a start Marker ID field, an End App ID field, an End Marker ID field, and a MarkerPair Name field.

Sub A5

16. A method for obtaining benchmark timing for a portion of an application program execution, the method comprising:

inserting one or more code markers into the application program at predefined locations within the application program corresponding to the point at which benchmark timing data is desired;

Sub A5 int

determining if benchmark timing data is to be collected at each code marker by checking for the existence of processing modules identified by an identification key within a system registry;

if benchmark timing data is to be collected at each code marker:

generating a benchmark data record containing the collected benchmark timing data each time the code markers are reached;

storing the benchmark data records within a data memory block within the processing modules identified by the identification key within the system registry;

retrieving the benchmark data records from the data memory block for transfer to first data record in a Raw Data Table device once all of the run-time internal state data has been collected; and

processing the first data records stored within the Raw Data Table to generate second data records in a Processed Data Table that estimate the benchmark timing defined between two benchmark data records.

Sub B4

17. The method according to claim 16, wherein the benchmark timing generated and stored within the processed data table is determined from difference between two benchmark timestamp data entries stored within the raw data table.

18. The method according to claim 17, wherein

the benchmark timing is determined by subtracting an estimate for the total overhead processing from the difference between two benchmark timestamp data entries stored within the raw data table;

the estimate for the total overhead processing is determined by totaling the difference between an overhead timestamp value and a benchmark timestamp value for all code markers between the two benchmark timestamp entries used to determine the benchmark timing;

the benchmark timestamp value is obtained from a system timer immediately after a code marker is reached; and

Sub B4 end

the overhead timestamp value is obtained from the system timer immediately before the processing returns to the application program from performance marker processing.

.19. The method according to claim 18, wherein the first data record within the Raw Data Table comprises a ResultID field, an AppID field, a MarkerID field, a Marker Cycles Field, and an Overhead Cycles field.

Sub A6

20. The method system according to claim 19, wherein the second data record of the Processed Data Field comprises a ResultID field, a Reboot Iteration field, a Launch Iteration field, a Marker Iteration field, a Marker Pair ID field, and a Seconds Field.

Sub B4

21. The method according to claim 22, wherein the ResultID field of the Raw Data Table corresponds to the ResultID field in the Processed Data Table.

22. The method according to claim 21, wherein the Marker Pair ID field corresponds to a second Marker Pair ID field in a Marker Pair Table.

23. The method according to claim 22, wherein the Marker Pair Table comprises the second Marker Pair ID field, a start App ID field, a start Marker ID field, an End App ID field, an End Marker ID field, and a MarkerPair Name field.

24. A computer data product readable by a computing system and encoding a computer program of instructions for executing a computer process for obtaining run-time internal state data within an application program, said computer process comprising:

Sub A7

inserting one or more code markers into the application program at predefined locations within the application program corresponding to the point at which benchmark timing data is desired;

determining if benchmark timing data is to be collected at each code marker by checking for a processing modules identified by an identification key within a system registry;

if benchmark timing data is to be collected at each code marker:

generating a benchmark data record containing the collected benchmark timing data each time the code markers are reached;

storing the benchmark data records within a data memory block within the processing modules identified by the identification key within the system registry;

retrieving the benchmark data records from the data memory block for transfer to first data record in a Raw Data Table device once all of the run-time internal state data has been collected; and

processing the first data records stored within the Raw Data Table to generate second data records in a Processed Data Table that estimate the benchmark timing defined between two benchmark data records;

wherein the benchmark timing generated and stored within the processed data table is determined from difference between two data entries stored within the raw data table.

Sub B8

25. The computer data product according to claim 24, wherein the benchmark timing is determined by subtracting an estimate for the total overhead processing from the difference between two benchmark timestamp data entries stored within the raw data table;

the estimate for the total overhead processing is determined by totaling the difference between an overhead timestamp value and a benchmark timestamp value for all code markers between the two benchmark timestamp entries used to determine the benchmark timing;

the benchmark timestamp value is obtained from a system timer immediately after a code marker is reached; and

the overhead timestamp value is obtained from the system timer immediately before the processing returns to the application program from performance marker processing.

Sub B8 Cont

26. The computer data product according to claim 25, wherein the first data record within the Raw Data Table comprises a ResultID field, an AppID field, a MarkerID field, a Marker Cycles Field, and an Overhead Cycles field.

27. The computer data product system according to claim 26, wherein the second data record of the Processed Data Field comprises a ResultID field, a Reboot Iteration field, a Launch Iteration field, a Marker Iteration field, a Marker Pair ID field, and a Seconds Field.

28. The computer data product according to claim 27, wherein the ResultID field of the Raw Data Table corresponds to the ResultID field in the Processed Data Table.

29. The computer data product according to claim 28, wherein the Marker Pair ID field corresponds to a second Marker Pair ID field in a Marker Pair Table.

30. The computer data product according to claim 29, wherein the Marker Pair Table comprises the second Marker Pair ID field, a start App ID field, a start Marker ID field, an End App ID field, an End Marker ID field, and a MarkerPair Name field.

Sub B8 cnd

31. The computer data product according to claim 23, wherein the computer data product comprises a computer readable storage medium readable by a computer upon which encoded instructions used to implement the computer process are stored.